

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的快递管理的数据库设计 | | |
| 课 程 | Oracle数据库应用 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2017级 |
| 学生姓名 | 张楠夫 | 学号 | 201710414428 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 10 |  |
| 表设计 | 表，表空间设计合理，数据合理 | 20 |  |
| 用户管理 | 权限及用户分配方案设计正确 | 10 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 25 |  |
| 备份方案 | 备份方案设计正确 | 25 |  |
| 容灾方案 | DataGuard设计正确 | 10 |  |
| **得分合计** | | |  |

2019 年 11 月 18 日

**目 录**

1.快递信息管理系统E-R模型

1.1实体模型.............................................3

1.2 实体联系模型.........................................4

2.数据表的设计......................................5

3.创建PDB及用户，空间分配

\*创建PDB...........................................6

\*空间分配..........................................6-8

1. 创建表和插入数据

\*创建表...........................................8-17

5.创建序列，视图

\*创建序列...........................................17

6.创建程序包

\*创建程序包.........................................17

7. 设计手动备份方案

\*手动备份方案......................................20

8.实验部分截图

\*实验部分截图......................................21

9.实验总结

\*实验总结.........................................21

1.快递信息管理系统E-R模型

本系统是模拟某快递公司对商品快递的管理过程。管理模式为：某收件人通过商家网站，或联系商家等方式，购买了商家里的某商品，告知商家个人信息。而商家则把商品交给快递公司，由快递公司的管理员发给商家一个有相关信息的单子，让商家填写相关信息，确认后返回给管理员，管理员再生成快递单，交由管理的快递员，即让快递员领取快递，并进行派送至目的地，等待客户收货。

1.1实体模型

实体有：管理员，商品和快递员

管理员（MANAGER）：包含管理员ID，账号ID，性别，联系电话，地址，见图1-1：

MANAGER(管理员)

图1-1 管理员（MANAGER）实体

商品（PRODUCTS）：包含商品ID，商品名称，商品类型，商家，价格，见图1-2：

PRODUCTS（商品）

图1-2 商品（PRODUCTS）实体

快递员（COURIER）：包含快递员ID，姓名，性别，联系电话，工作状态，工资，入职日期，见图1-3：

COURIER（快递员）

图1-3 快递员（COURIER）实体

1.2 实体联系模型

快递员的工作就是派送商品到顾客手中，快递员与商品之间就有一个派送的关系，见图1-4有快递员与商品的关系是一对多的关系：

1

M

派送

快递员

商品

图1-4

再把派送关系细分为快递单和快递详单两个实体，快递单中存储：快递单ID（PACKAGE\_ID），发货方（SELLER），收货人信息（RECEIVER \_NAME, RECEIVER \_TEL, RECEIVER \_ADDRESS），快递公司信息（COMPANY,COMPANY\_TEL），快递单生成日期（PACKAGE\_DATE），而快递详单中有：详单编号（ID），商品数量（PRODUCT\_NUM）,商品总价格（PRODUCT\_PRICE），见图1-5：

M

1

M

快递详单

1

1

商品

快递单

快递员

M

图1-5

2.数据表的设计

管理员表（MANAGER）：见表2-1

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 注释 |
| MANAGER\_ID | NUMBER(6,0) | NO | 管理员ID，主键 |
| MANAGER\_NAME | VARCHAR(20BYTE) | NO | 管理员姓名 |
| SEX | VARCHAR2(4BYTE) | NO | 管理员性别，如：男性 |
| TEL | VARCHAR2(40BYTE) | NO | 管理员电话，11位 |
| WORKPLACE | VARCHAR2(40BYTE) | NO | 管理员办公室 |

表2-1 管理员表

商品表（PRODUCTS）：见表2-2

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 注释 |
| PRODUCTS\_ID | VARCHAR2(40BYTE) | NO | 商品ID，主键 |
| PRODUCTS\_NAME | VARCHAR2(40BYTE) | NO | 商品名称 |
| PRODUCTS\_TYPE | VARCHAR2(40BYTE) | NO | 商品类型，食品,服装,美妆,手机 |

表2-2 商品表

快递员表（COURIER）：见表2-3

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 注释 |
| COURIER\_ID | NUMBER(6,0) | NO | 快递员ID，主键 |
| COURIER\_NAME | VARCHAR2(40BYTE) | NO | 快递员姓名 |
| COURIER\_SEX | VARCHAR2(4BYTE) | NO | 快递员性别，如：男性 |
| COURIER\_TEL | VARCHAR2(40BYTE) | YES | 快递员联系电话 |
| STATUS | VARCHAR2(6BYTE) | NO | 快递员工作状态：空闲,派送中 |
| SALARY | NUMBER(8,2) | YES | 工资，要大于等于0 |
| HIRE\_DATE | DATE | NO | 入职日期 |
| MANAGER\_ID | NUMBER(6,0) | YES | 管理员的外键 |

表2-3 快递员表

快递单表（PACKAGE）：见表2-4

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 注释 |
| PACKAGE\_ID | NUMBER(10,0) | NO | 快递单ID，主键，值来自序列：SEQ\_PACKAGE\_ID |
| SALLER | VARCHAR2(40BYTE) | NO | 发送方，即商家名称 |
| RECEIVER\_NAME | VARCHAR2(40BYTE) | NO | 收货人姓名 |
| RECEIVER \_TEL | VARCHAR2(40BYTE) | NO | 收货人联系电话 |
| RECEIVER \_ADDRESS | VARCHAR2(40BYTE) | NO | 收货人的收货地址 |
| COMPANY | VARCHAR2(40BYTE) | NO | 快递公司名称 |
| COMPANY\_TEL | VARCHAR2(40BYTE) | NO | 快递公司电话 |
| PACKAGE\_DATE | DATE | NO | 快递单生成日期 |
| COURIER\_ID | NUMBER(6,0) | NO | 快递员ID，快递员表的外键 |

表2-4快递单表

快递详单表（PACKAGE\_DETAILS）：见表2-5

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 注释 |
| DETAILS\_ID | NUMBER(10,0) | NO | 本表主键，值来自序列：SEQ\_PACKAGE\_DETAIL\_ID |
| PACKAGE\_ID | NUMBER(10,0) | NO | 快递单表的外键 |
| PRODUCTS\_ID | VARCHAR(40BYTE) | NO | 商品表外键 |
| PRODUCTS\_NUM | NUMBER(8,2) | NO | 商品数量 |
| PRODUCTS\_PRICES | NUMBER(8,2) | NO | 商品的总价值 |

在上述表中，快递员表的MANAGER\_ID属性表示此快递员会被某个管理员指派去派送快递，即是快递员的上司，而当删除时此属性置为空即可。而快递详单表中的PRODUCTS\_PRICES属性即为该商品价格的总和，是商家提供的，不需要知道该商品的单价，即不需要自己再去计算。

3.创建PDB及用户，空间分配

在系统终端上传建一个pdbmyself的可插接式数据库PDB，其中不包含任何数据，是空的，

打开一个系统终端，操作语句语句为：

$ sqlplus / as sysdba

SQL> create pluggable database pdbmysef

admin user user1 identified by 123 file\_name\_convert=

(‘/home/oracle/app/oracle/oradata/orcl/pdbseed/’

,’/home/oracle/app/oracle/oradata/orcl/pdbmyself/’)；

//为了排版好看，区分明确，代码中间有多余空格，使代码不显得拥挤

该PDB创建的同时，也创建了用户“user1”，此用户没有分配任何东西。然后在SQL Developer中新建连接，用户名填写系统“system”，而服务名则填写“pdbmysef”，即刚刚新建的PDB名称，密码都是“123”，输入后，还不能连接刚刚创建的数据库pdbmysef，因为这个数据库还没有被启动，启动方法有两种，此处就选择较直接的一种：

SQL> STARTUP PLUGGABLE DATABASE PDBMYSEF;

Pluggable Databse opened.

出现上述结果后，就可以成功连接到数据库，接下来的大部分操作就可以在SQL Developer中操作了。

然后再在system用户下创建两个表空间user01（分配一个数据文件：pdbmyself\_user01.dbf，大小为100M）和user02（分配两个数据文件：pdbmyself\_user02\_1.dbf和pdbmyself\_user02\_2.dbf，大小各自为100M，共计200M）。

Create tablespace user01 datafile

‘/home/oracle/app/oracle/oradata/orcl/pdbmyself/pdbmyself\_user01.dbf’

size 100M autoextend on next 50M maxsize unlimited extent management

local segment space management auto;

Create tablespace user02 datafile

‘/home/oracle/app/oracle/oradata/orcl/pdbmyself/pdbmyself\_user02\_1.dbf’

size 100M autoextend on next 50M maxsize unlimited extent management

local segment space management auto;

Create tablespace user02 datafile

‘/home/oracle/app/oracle/oradata/orcl/pdbmyself/pdbmyself\_user02\_2.dbf’

size 100M autoextend on next 50M maxsize unlimited extent management

local segment space management auto;

此处再列出将要创建的两个角色（con\_res\_view,user1\_view）以及给不同时间创建的两个用户（user1,student）的权限有哪些：

|  |  |  |
| --- | --- | --- |
| 用户 | 角色 | 权限 |
| student（对两个表空间的权限无限制） | con\_res\_view | connect,可以连接数据库的权限 |
| resource,可以创建资源（表，过程，序列等资源对象） |
| create view,可以创建视图 |
| select on 五张表，可以查询五张表 |
| user1 | user1\_view | Select on view\_package\_details,可以查询视图 |

表空间user01和user02创建完成之后，就可以再创建一个用户“student”，为此用户分配表空间的使用配额，分配角色CONNECT和RESOURCE，以便用户可以连接数据库，可以创建资源（表，过程，序列等资源对象），还可以为其分配一个系统权限：创建视图（create view）。

create user student identified by 123

default tablespace “users01”

temporary tablespace “temp”;

--quotas

alter user student quota unlimited on users01;

alter user student quota unlimited on users02;

--roles

create role con\_res\_view;

grant connect，resource，create view to con\_res\_view;

grant con\_res\_view to student;

而用户user1则只分配权限“select on view\_package\_details”,即后续会创建的快递详单视图，此用户只会查询此详单视图的信息，无法修改，增添或者删除，具体的分配权限操作会在后面进行。

至此，已完成空间表，角色和用户的创建已完成。而根据表的张数及会有五万条数据，所以会把PACKAGE和PACKAGE\_DETAILS两张表设计为基于PACKAGE\_DATE的分区表，以加快局部时间范围的查询速度，存储空间的分配规划表如3-1：

|  |  |  |
| --- | --- | --- |
| 表 | 表空间user01 | 表空间user02 |
| MANAGER | 存储全部数据 |  |
| PRODUCTS | 存储全部数据 |  |
| CURIER | 存储全部数据 |  |
| PACKAGE | 存储2017年之前（不含）的数据 | 存储2017年之后（含）的数据 |
| PACKAGE\_DETAILS | 存储2017年之前（不含）的数据 | 存储2017年之后（含）的数据 |

表3-1

4.创建表和插入数据

在系统用户system上创建管理员表（MANAGER）：

CREATE TABLE manager

(

manager\_id NUMBER(6,0) NOT NULL，

manager\_name VARCHAR2(20)NOT NULL，

Sex VARCHAR2(4) NOT NULL,

TEL VARCHAR2(40)NOT NULL,

WORKPLACE VARCHAR2(40)NOT NULL,

CONSTRAINT "MANAGER\_PK" PRIMARY KEY (manager\_id)

)PCTFREE 10 PCTUSED 40 INITRANS 1

MAXTRANS 255

LOGGING

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

BUFFER\_POOL DEFAULT

)

TABLESPACE USERS01;

创建商品表（PRODUCTS）：

CREATE TABLE products

(

products\_id VARCHAR2(640) NOT NULL，

products\_name VARCHAR2(40)NOT NULL，

products\_type VARCHAR2(40) NOT NULL,

CONSTRAINT "PRODUCTS\_PK" PRIMARY KEY (products\_id)

)

PCTFREE 10

INITRANS 1

LOGGING

STORAGE

(

BUFFER\_POOL DEFAULT

)

TABLESPACE USERS01;

创建快递员表（COURIER）：

CREATE TABLE courier

(

courier\_id NUMBER(6,0) NOT NULL，

courier\_name VARCHAR2(40)NOT NULL，

courier\_sex VARCHAR2(4) NOT NULL,

courier\_tel VARCHAR2(40) NOT NULL,

status VARCHAR2(6) NOT NULL,

salary VARCHAR2(40) NOT NULL,

hire\_date VARCHAR2(40) NOT NULL,

manager\_id NUMBER(6,0) NOT NULL,

CONSTRAINT "COURIER\_PK" PRIMARY KEY (courier\_id),

CONSTRAINT "COURIER\_FK" FOREIGN KEY (manager\_id) REFERENCES MANAGER(manager\_id)

)

PCTFREE 10

PCTUSED 40

INITRANS 1

MAXTRANS 255

LOGGING

STORAGE

(

INITIAL 65536

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS 2147483645

BUFFER\_POOL DEFAULT

)

TABLESPACE USERS01;

创建快递单表（PACKAGE）：

CREATE TABLE package

(

package\_id NUMBER(10, 0) NOT NULL,

saller VARCHAR2(40) NOT NULL,

receiver\_name VARCHAR2(40) NOT NULL,

receiver\_tel VARCHAR2(40) NOT NULL,

receiver\_address VARCHAR2(40) NOT NULL,

company VARCHAR2(40) NOT NULL,

company\_tel VARCHAR(40) NOT NULL,

package\_date DATE NOT NULL,

courier\_id NUMBER(6, 0) NOT NULL,

CONSTRAINT PACKAGE\_PK PRIMARY KEY(package\_id)

USING INDEX

(

CREATE UNIQUE INDEX PACKAGE\_PK ON PACKAGE (PACKAGE\_ID ASC)

LOGGING

TABLESPACE USERS01

PCTFREE 10

INITRANS 2

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOPARALLEL

),

CONSTRAINT "PACKAGE\_FK" FOREIGN KEY (courier\_id) REFERENCES COURIER(courier\_id)

ENABLE

)

TABLESPACE USERS01

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS

NOPARALLEL

PARTITION BY RANGE (PACKAGE\_DATE)

(

PARTITION PARTITION\_2015 VALUES LESS THAN (TO\_DATE(' 2016-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS01

PCTFREE 10

INITRANS 1

STORAGE

(

INITIAL 8388608

NEXT 1048576

MINEXTENTS 1

MAXEXTENTS UNLIMITED

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_2016 VALUES LESS THAN (TO\_DATE(' 2017-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS01

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_2017 VALUES LESS THAN (TO\_DATE(' 2018-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS01

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_2018 VALUES LESS THAN (TO\_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS02

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_2019 VALUES LESS THAN (TO\_DATE(' 2020-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS02

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

, PARTITION PARTITION\_2020 VALUES LESS THAN (TO\_DATE(' 2021-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))

NOLOGGING

TABLESPACE USERS02

PCTFREE 10

INITRANS 1

STORAGE

(

BUFFER\_POOL DEFAULT

)

NOCOMPRESS NO INMEMORY

);

创建快递详单表（PACKAGE\_DETAILS）：

CREATE TABLE package\_details

(

details\_id NUMBER(10, 0) NOT NULL

, package\_id NUMBER(10, 0) NOT NULL

,product\_name VARCHAR2(40) NOT NULL

, product\_num NUMBER(8, 2) NOT NULL

, product\_prices NUMBER(8, 2) NOT NULL

, CONSTRAINT PACKAGE\_DETAILS\_PK PRIMARY KEY (id)

,CONSTRAINT PACKAGE\_DETAILS\_FK1 FOREIGN KEY (package\_id)

REFERENCES package ( package\_id )

,CONSTRAINT PACKAGE\_DETAILS\_FK2 FOREIGN KEY (products\_id)

REFERENCES products ( products\_id )

ENABLE

)

TABLESPACE USERS01

PCTFREE 10 INITRANS 1

STORAGE (BUFFER\_POOL DEFAULT )

NOCOMPRESS NOPARALLEL

PARTITION BY REFERENCE (package\_details\_fk1);

分别对五张表插入数据，管理员表（MANAGER）其中有五条数据，即有五个管理员的信息存入此表中：

INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201611,'zhanghong','MAN','13890011001','p\_100');

INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201712,'zhanglin','MAN','13890011002','p\_120');

INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201813,'zhangxiu','MAN','13890011003','p\_120');

INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201914,'zhangmei','GIRL','13890011004','p\_100');

INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201715,'zhanghao','MAN','13890011005','p\_120');

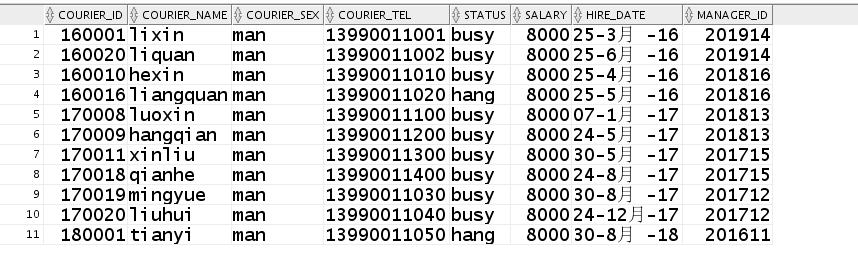
INSERT INTO MANAGER(MANAGER\_ID,MANAGER\_NAME,SEX,TEL,WORKPLACE)

VALUES(201816,'zhanghe','MAN','13890011006','p\_120');

商品表(PRODUCTS)共16条数据

|  |  |  |
| --- | --- | --- |
| PRODUCTS\_ID | PRODUCTS\_NAME | PRODUCTS\_TYPE |
| food01 | quqi cookies | food |
| food02 | there squirrels | food |
| food03 | ambpoeial | food |
| food04 | leys | food |
| clothes01 | U coat | clothes |
| clothes02 | U jacket | clothes |
| clorhes03 | U skirt | clothes |
| clothes04 | U dress | clothes |
| face01 | guuci | face |
| face02 | +wis+ | face |
| face03 | MAN COODS | face |
| face04 | gu long | face |
| phone01 | meizu pro6s | phone |
| phone02 | oppo renew | phone |
| Phone03 | iphone11 | phone |
| phine04 | HUAWEI Mate10 | phone |

快递员表（COURIER）共11条数据,查询表可得截图如下:



快递单表（PACKAGE）共计一万条数据，快递详单（PACKAGE\_DETAILS）相应地会有四万条数据，以下是插入数据的脚本文件。具体含义是：先声明要插入数据库表的数据的变量，然后再利用for循环改变变量的值，在赋值进入插入语句中的值参数。这样的数据为模拟数据，

不具有逻辑性和真实性，只具有测试作用。快递单的生成的日期是间隔的的除了2015年和2020年，中间的年份都包含月份1,11,12月三月的快递单。快递单的派送人员由11位快递员

循环派送，其他的快递单表中的某些信息则可以固定，不用变量使其改变。而快递详单表的

信息对应一个快递单有四个快递详单表，即一次派送四件商品，包含商品表中四类商品，每类一样，共十六样，然后再循环使用数据，再根据商品类别，调整随机函数给出一个商品的总价格赋值给变量，具体代码如下：

declare

v\_package\_id number(10);

v\_saller varchar2(40);

r\_name varchar2(40);

r\_tel varchar2(40);

r\_address varchar2(40);

v\_package\_dt date;

v\_courier\_id NUMBER(6);

v\_detail\_id number(10);

v\_products\_id varchar2(40);

v\_prices number(8,2);

begin

v\_detail\_id:=1;

delete from package\_details;

delete from package;

for i in 1..10000

loop

if i mod 6 =0 then

v\_package\_dt:=to\_date('2015-11-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60); --PARTITION\_2015-2016

elsif i mod 6 =1 then

v\_package\_dt:=to\_date('2016-11-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60); --PARTITION\_2016-2017

elsif i mod 6 =2 then

v\_package\_dt:=to\_date('2017-11-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60); --PARTITION\_2017-2018

elsif i mod 6 =3 then

v\_package\_dt:=to\_date('2018-11-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60); --PARTITION\_2018-2019

elsif i mod 6 =4 then

v\_package\_dt:=to\_date('2019-11-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60); --PARTITION\_2019-2020

else

v\_package\_dt:=to\_date('2020-9-11 00:00:00','yyyy-mm-dd hh24:mi:ss')+(i mod 60);

--PARTITION\_2020

end if;

v\_courier\_id:=CASE I MOD 11 WHEN 0 THEN 160001 WHEN 1 THEN 160020

WHEN 2 THEN 160010 WHEN 3 THEN 160016

WHEN 4 THEN 170008 WHEN 5 THEN 170009

WHEN 6 THEN 170011 WHEN 7 THEN 170018

WHEN 8 THEN 170019 WHEN 9 THEN 170020

ELSE 180001 END;

--插入订单

v\_package\_id:=i;

v\_saller:='company';

r\_name:= 'zhang' || i;

r\_tel:= '139888883' || i;

r\_address:='P-C-'|| i mod 10;

Insert into PACKAG(PACKAGE\_ID,SALLER,RECEIVER\_NAME,RECEIVER\_TEL,

RECEIVER\_ADDRESS,COMPANY,COMPANY\_TEL,PACKAGE\_DATE,COURIER\_ID)valus

(v\_package\_id,v\_saller,r\_name,r\_tel,r\_address,'ZTFast','12306',v\_package\_dt,v\_courier\_id);

--插入订单y一个订单包括4个产品

v\_prices:=dbms\_random.value(10,400);

v\_products\_id:='food0'|| (i mod 4 + 1);

insert into PACKAGE\_DETAILS(DETAILS\_ID,PACKAGE\_ID,PRODUCTS\_ID,PRODUCTS\_NUM,PRODUCTS\_PRICES)

values (v\_detail\_id,v\_package\_id,v\_products\_id,2,v\_prices);

v\_prices:=dbms\_random.value(500,1500);

v\_products\_id:='clothes0'|| (i mod 4 + 1);

v\_detail\_id:=v\_detail\_id+1;

insert into PACKAGE\_DETAILS(DETAILS\_ID,PACKAGE\_ID,PRODUCTS\_ID,PRODUCTS\_NUM,PRODUCTS\_PRICES)values (v\_detail\_id,v\_package\_id,v\_products\_id,3,v\_prices);

v\_prices:=dbms\_random.value(1000,2000);

v\_products\_id:='face0'|| (i mod 4 + 1);

v\_detail\_id:=v\_detail\_id+1;

insert into PACKAGE\_DETAILS(DETAILS\_ID,PACKAGE\_ID,PRODUCTS\_ID,PRODUCTS\_NUM,PRODUCTS\_PRICES)

values (v\_detail\_id,v\_package\_id,v\_products\_id,1,v\_prices);

v\_prices:=dbms\_random.value(2000,9000);

v\_products\_id:='phone0'|| (i mod 4 + 1);

v\_detail\_id:=v\_detail\_id+1;

insert into PACKAGE\_DETAILS(DETAILS\_ID,PACKAGE\_ID,PRODUCTS\_ID,PRODUCTS\_NUM,PRODUCTS\_PRICES)

values (v\_detail\_id,v\_package\_id,v\_products\_id,1,v\_prices);

v\_detail\_id:=v\_detail\_id+1;

IF I MOD 1000 =0 THEN

commit; --每次提交会加快插入数据的速度

END IF;

end loop;

end;

/

5.创建序列，视图

在插入快递单表的数据的时候，需要用到自增的ID，所以需要设计两个序列。SEQ\_PACKAGE\_

ID用于给PACKAGE表的package\_id赋值，SEQ\_PACKAGE\_DETAILS\_ID用于给PACKAGE\_DETAILS

表赋值。序列·是这样的，取值从1开始，最大值取到9999999999，每次增加1，存放序列块大小为2000BYTE,序列值达到最大值后不循环，不重新划分内存块,具体代码如下：

CREATE SEQUENCE “SEQ\_PACKAGE\_ID” MINVALUE 1

MAXVALUE 9999999999 INCREMENT BY 1

START WITH 1 CACHE 2000 ORDER NOCYCLE NOPARTITION；

CREATE SEQUENCE “SEQ\_PACKAGE\_DETAILS\_ID” MINVALUE 1

MAXVALUE 9999999999 INCREMENT BY 1

START WITH 1 CACHE 2000 ORDER NOCYCLE NOPARTITION;

为了方便地查询快递详单表中的快递单，需要设计一个视图VIEW\_PACKAGE\_DETAILS，包含快递详单的id，快递单id，收件人的姓名，联系电话，收件地址商品id，商品名称，商品数量，商品总价格，这个视图的代码如下：

CREATE OR REPLACE VIEW VIEW\_PACKAGE\_DETAILS

AS SELECT

p.DETAILS\_ID, a.PACKAGE\_ID, a.RECEIVER\_NAME, a.RECEIVER\_TEL, a.RECEIVER\_ADDRESS

p.PRODUCTS\_ID, c.PRODUCTS\_NAME, c.PRODUCTS\_TYPE, p.PRODUCTS\_NUM,p.PRODU

CTS\_PRICES FROM PACKAGE a, PACKAGE\_DETAILS p, PRODUCTS c

WHERE p.PACKAGE\_ID=a.PACKAGE\_ID AND p.PRODUCTS\_ID=c.PRODUCTS\_ID;

此时还应将select on VIEW\_PACKAGE\_DETAILS的权限赋给用户user1，student，需要先赋给角色con\_res\_view,然后再创建角色user1\_view，赋给他查看此视图的权限：

grant select on view\_package\_details to con\_res\_view;

grant con\_res\_view to student;

create role user1\_view;

grant connect,select on view\_package\_details to user1\_view;

grant user1\_view to user1;

6.创建程序包

创建一个程序包，包名为MyPack，此程序包中有一个函数和三个过程，分别为：

函数一，查询某产品的快递单总数:Get\_ProductNum(V\_PRODUCTS\_ID VARCHAR2(40)),

过程一，查询某个区的所有快递单的信息:Get\_AddressPackage(V\_KEY\_ADDRESS VARCHAR2(40)),

过程二，查询在某个管理员下的快递员派送的总单子数:Get\_AllPackage(V\_MANAGER\_ID NUMBER),

过程三，查询某个商品的快递总数量和总金额:Get\_NumAndPrice(V\_PRODUCTS\_ID VARCHAR2(40)).

在用户system下创建，再分别用模拟数据，调用函数和过程测试是否成功创建：

create or replace PACKAGE MyPack IS

FUNCTION Get\_ProductNum(V\_PRODUCTS\_ID VARCHAR2) RETURN NUMBER;

PROCEDURE Get\_AddressPackage(V\_KEY\_ADDRESS VARCHAR2);

PROCEDURE Get\_AllPackage(V\_MANAGER\_ID NUMBER);

PROCEDURE Get\_NumAndPrice(V\_PRODUCTS\_ID VARCHAR2);

END MyPack;

create or replace PACKAGE BODY MyPack IS

FUNCTION Get\_ProductNum(V\_PRODUCTS\_ID VARCHAR2) RETURN NUMBER

AS

N NUMBER(20,2);

BEGIN

SELECT COUNT(\*) into N FROM PACKAGE\_DETAILS O

WHERE O.PRODUCTS\_ID=V\_PRODUCTS\_ID;

RETURN N;

END;

PROCEDURE Get\_AddressPackage(V\_KEY\_ADDRESS VARCHAR2)

AS

begin

for v in

(SELECT package\_id,receiver\_address FROM package

WHERE receiver\_address like '%'||V\_KEY\_ADDRESS)

LOOP

DBMS\_OUTPUT.PUT\_LINE(' '||V.PACKAGE\_ID||' '||v. receiver\_address);

END LOOP;

END;

PROCEDURE Get\_AllPackage(V\_MANAGER\_ID NUMBER)

AS

begin

--使用游标

for v in

(SELECT COURIER\_ID,COUNT(COURIER\_ID)AS Total FROM PACKAGE WHERE COURIER\_ID IN(SELECT COURIER\_ID FROM COURIER WHERE MANAGER\_ID=V\_MANAGER\_ID))

LOOP

DBMS\_OUTPUT.PUT\_LINE(V\_MANAGER\_ID||‘ ‘||V.COURIER\_ID||' '||v.Total);

END LOOP;

END;

PROCEDURE Get\_NumAndPrice(V\_PRODUCTS\_ID VARCHAR2)

AS

Begin

For v in

(SELECT COUNT(PRODUCTS\_ID)AS Num,SUM(PRODUCTS\_PRICES)AS AllPrices FROM

PACKAGE\_DETAILS WHERE PRODUCTS\_ID=V\_PRODUCTS\_ID)

LOOP

DBMS\_OUTPUT.PUT\_LINE(V\_PRODUCTS\_ID||’ ’||v.Num||’ ‘||v.AllPrices);

END LOOP;

END;

END MyPack;

/

拟定测试调用函数的输入参数V\_PRODUCTS\_ID为‘food01’和’phone02’，即查询商品id为‘food01’，‘phone02’的商品的快递详单总数目：

Select MyPack.Get\_ProductNum(‘food01’)AS 商品quqicookies数量,MyPack.Get\_ProductNu

m(‘phone02’)AS 华为mate9 from dual;

拟定测试调用过程的输入参数V\_KEY\_ADDRESS，V\_MANAGER\_ID，V\_PRODUCTS\_ID分别为：

‘2’，201712，‘phone02’，具体操作如下：

set serveroutput on

DECLARE

V\_1 VARCHAR2(40);

V\_2 NUMBER;

V\_3 VARCHAR2(40);

BEGIN

V\_1:=’ 2‘;

V\_2:=201712;

V\_3:=’phone02’;

MyPack.Get\_AddressPackage(V\_KEY\_ADDRESS => V\_1);

MyPack.Get\_AllPackage(V\_MANAGER\_ID => V\_2);

MyPack.Get\_NumAndPrice(V\_PRODUCTS\_ID => V\_3);

END;

/

V\_1=’2’ 即是在区域P-C-2的快递单有哪些，请列出来，

V\_2:=201712 ，在管理员id为201712下的快递员派出的总件数是多少，

V\_3:=’phone02’ ，派出的快递中，商品id为‘phone02’的总价值为多少，即把快递详单中商品id为此值的总价加起来。

7. 设计手动备份方案

此方案是按照书P347上的13.4.6 实用案例：完全恢复一个PDB，根据此例子写出方案。方案的备份策略采用完整的压缩备份集方式备份，并且备份完成之后，删除过期的备份文件和归档日志文件，这个样做可以节约出db\_recovery\_file\_dest\_size允许的空间，数据库运行期间就不会产生归档日志满的错误。

恢复是恢复一个PDB数据库，即是创建的数据库pdbmysef。这样做的好处是在恢复时只需要使损坏的PDB停机，而不会将整个数据库停机，保证其他数据库正常使用。具体操作如下：

首先，在数据库pdbmysef正常工作，即数据还没有遭到破坏时，手动备份整个数据库。操作为：RMAN连接数据库，然后完整压缩备份整个数据库，删除过期的备份和归档日志，查看备份结果，主要看新创建的PDB数据库pdbmysef的备份集：

$ rman target /

RMAN> SHOW ALL;

...

RMAN> BACKUP AS COMPRESSED BACKUPSET DATABASE;

RMAN> LIST BACKUP;

RMAN> REPORT OBSOLETE;

RMAN> DELETE OBSOLETE;

如果在平时的操作过程中造成pdbmysef出故障，不能正常工作时需要对pdbmysef进行完全恢复，但在恢复之前需要先删除出故障的pdbmysef的所有文件，然后再关闭pdbmysef，在sqlplus中完成，重新打开一个系统命令窗口，运行sqlplus：

$ rman target /

RMAN> host “rm /home/oracle/app/oracle/oradata/orcl/pdbmysef/\*.dbf”;

$ sqlplus / as sysdba

SQL> SHOW PDBS;

...

SQL> ALTER SESSION SET CONTAINER =pdbmysef;

...

SQL> SHUTDOWN IMMEDIATE;

...

SQL> EXIT

上述操作中，show pdbs后虽然数据库文件全被删除了，但还是可以看到pdbmysef的状态任为正常的读写状态（READ\_WRITE），但是当session切换到pdbmysef里，执行各SQL语句时会发现“无法打开文件”的错误。最后之执行“showdown immediate;”看到“Pluggable Database closed”的时候，就表示pdbmysef被关闭了。此时就可以回到RMAN中单独恢复pdbmysef：

RMAN> RESTORE PLUGGABLE DATABASE pdbmysef；

RMAN> RECOVER PLUGGABLE DATABASE pdbmysef；

...

RMAN> ALTER PLUGGABLE DATABSE pdbmysef open;

最后的“ALTER PLUGGABLE DATABSE pdbmysef open;”命令打开pdbmysef没有报错，也没有加

Resetlogs选项，这就说明pdbmysef完全恢复成功，没有损失数据。

8. 部分实验截图

三个实体的数据表MANAGER,PRODUCTS,COURIER,通过查询语句分别可以得到：图8-1，

图8-2，图8-3

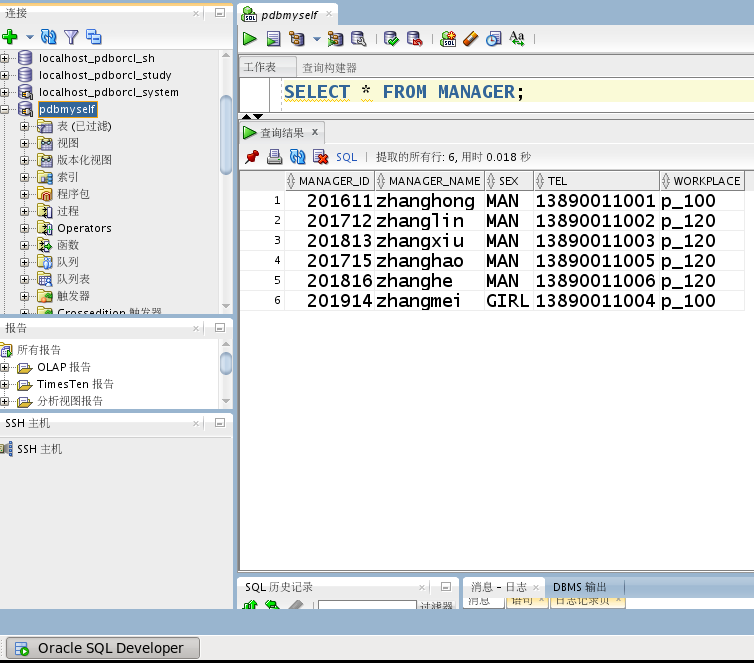


图8-1：MANAGER表

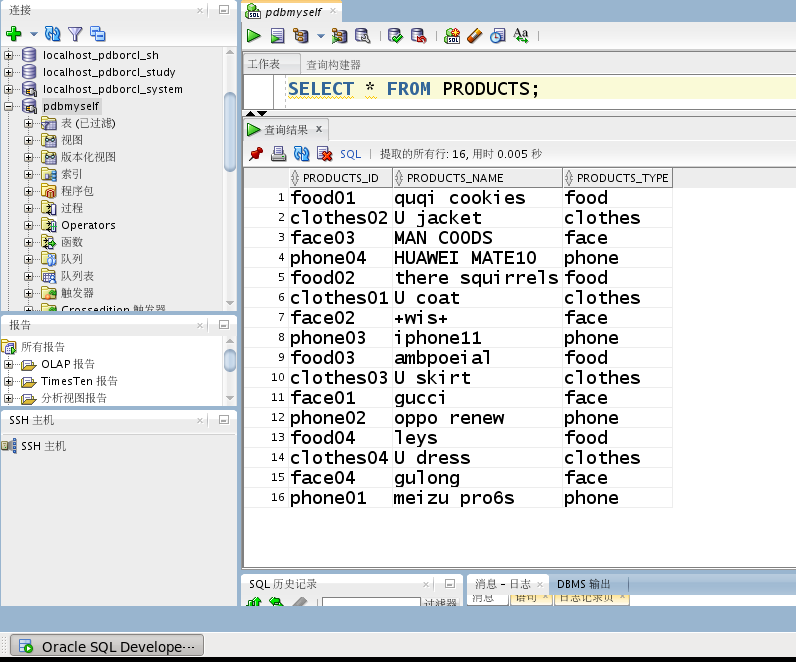


图8-2 PRODUCTS表

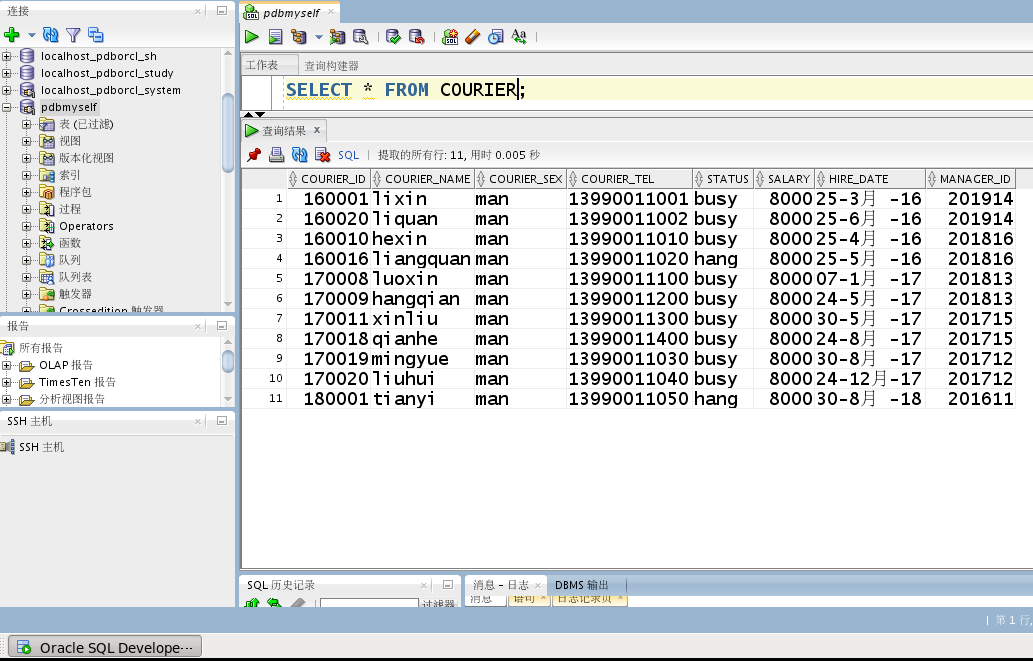


图8-3 COURIER表

由于PACKAGE，PACKAGE\_DETAILS表数据过多，可查看其数据条数：图8-4

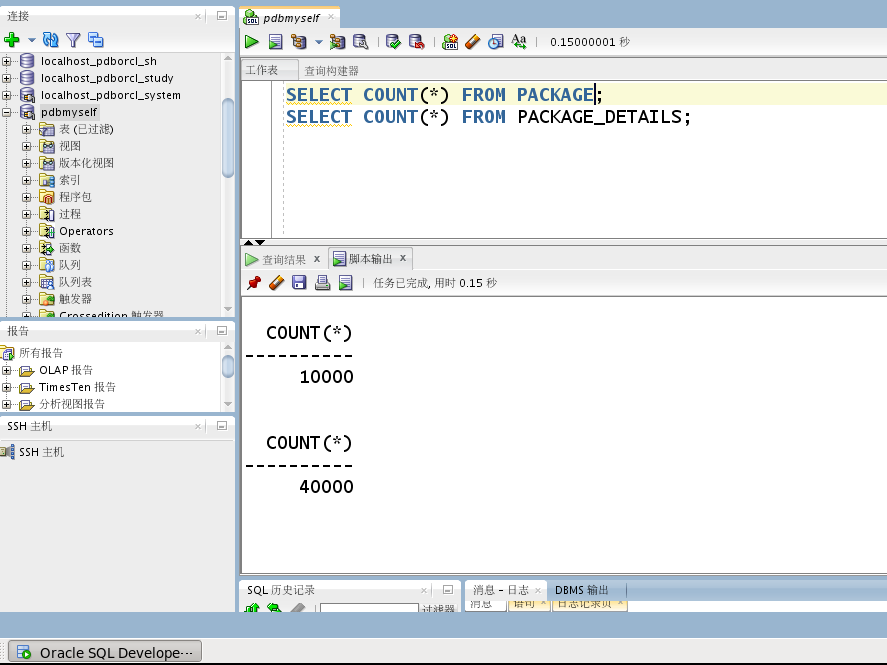


图8-4 五万条数据

创建视图VIEW\_PACKAGE\_DETAILS：图8-5

一个函数：图8-6

三个过程：图8-7

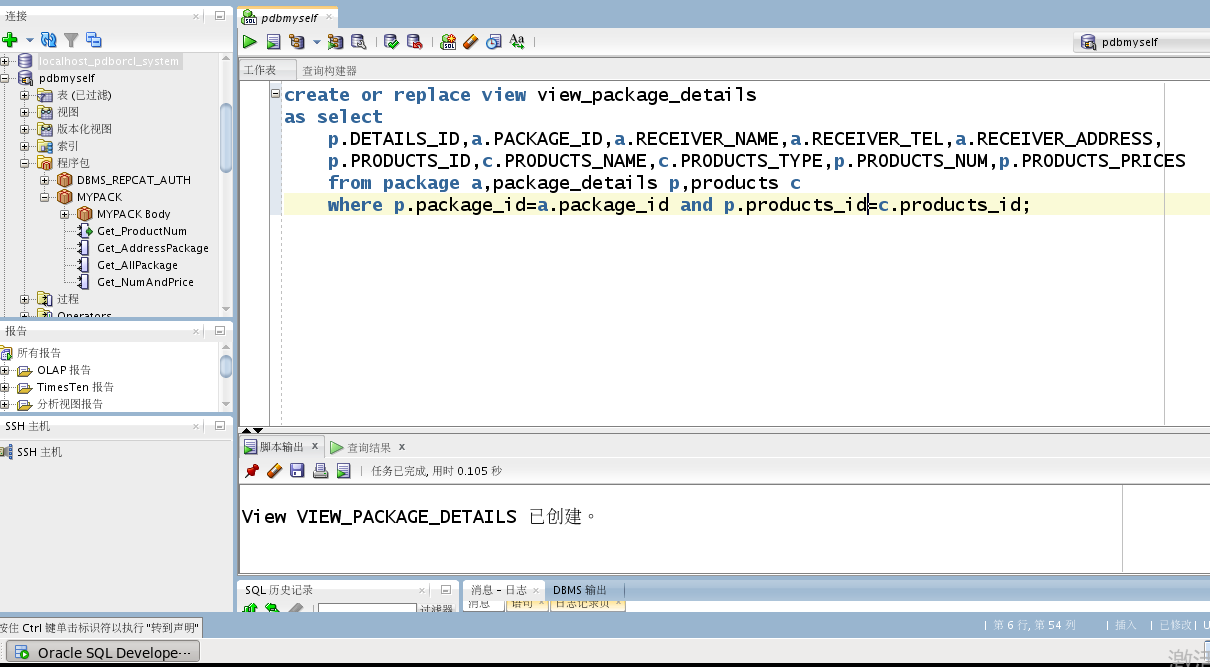


图8-5 视图VIWE\_PACKAGE\_DETAILS

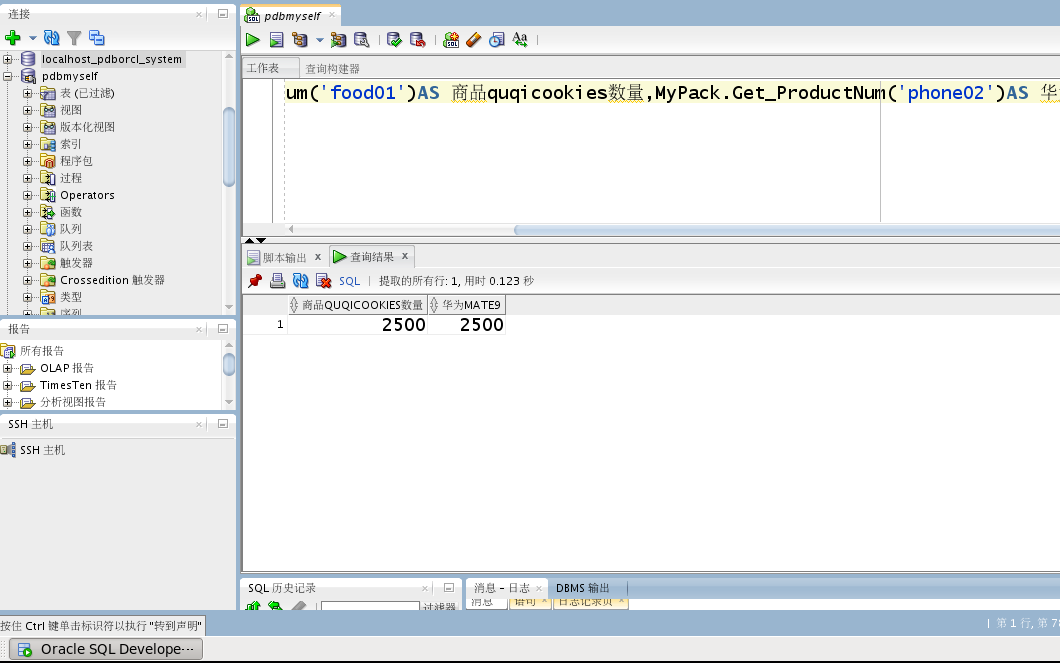


图8-6 一个函数

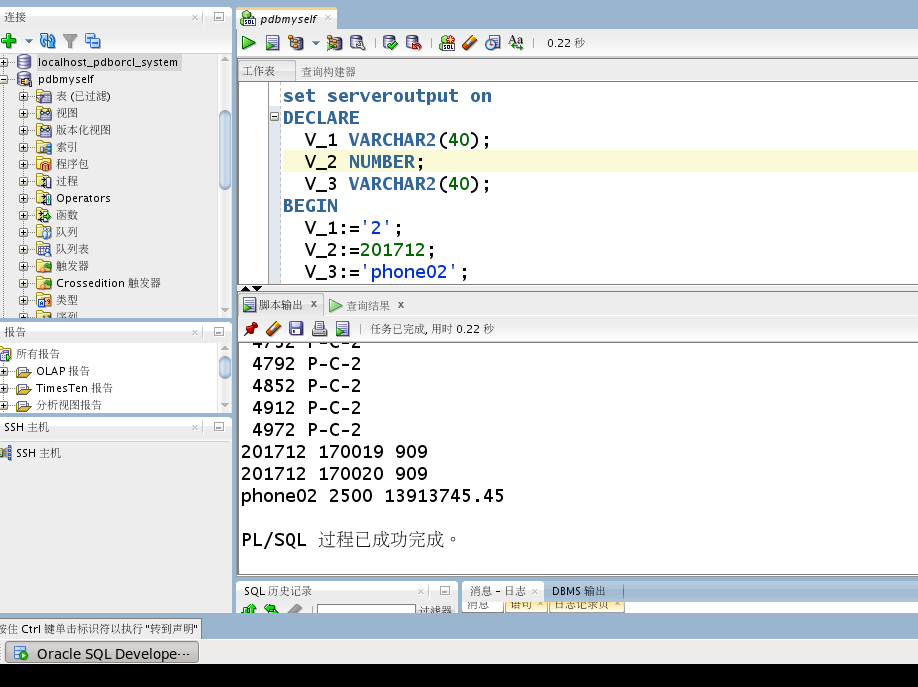


图8-7 三个过程

9. 期末考核总结

Oracle 数据库系统是美国 Oracle 公司(甲骨文)提供的以分布式数据库为核心的一组软件产品，是目前最流行的客户/服务器(CLIENT/SERVER)或 B/S 体系结构的数据库之一，是当前最流行的大型关系数据库之一，拥有广泛的用户和大量的应用案例。Oracle 数据库管理与开发已经成为一门综合性高、实践性强、应用领域广的技术类课程。对于从事数据维护和计算机程序开发的人员，在当前互联网时代，掌握 Oracle 数据库管理与开发是非常必要的。

通过这学期的学习过程：安装Oracle数据库，了解数据库特性，配置，创建与管理数据库，然后就是作业任务表空间，数据文件的维护，用户管理与共享对象训练，再然后就是表与数据的维护，最后讲解的是SQL语句，PL/SQL编程，备份与后恢复。这些流程都在五个平时实验中有所体现，通过书上的项目：小型商品销售系统，从头至尾完整学习和掌握 Oracle 数据库的开发过程和实际应用项目开发过程。而在老师给的服务器上进行实验操作，每个实验的任务都很明确，都体现了每个章节的内容。虽然没自己的电脑上安装Oracle数据库，缺乏安装的经验，对数据库了解程度还不够，但在后续实验中，通过老师的讲解以及书本的知识提示，还是能够掌握基础操作。

通过做此次期末实验设计，有几点感想，归纳如下：

1.扎实的理论基础。如果不掌握它们，很难写出高水平的程序.而这一点又是我们所缺乏的。

2.不钻牛角尖。当遇到障碍的时候，暂时远离电脑，当重新开始工作的时候，也许会发现那些难题现在竟然可以迎刃而解。

3.多与别人交流。三人行必有我师，从老师，同学，甚至是电脑里，都可以学到很多东西，还有可能会解决一些难题。

4.良好的编程风格。注意养成良好的习惯，代码的缩进编排，变量的命名规则要始终保持一致.如果注释和代码不一致，那就更加糟糕。

我深深的认识到，如果仅仅只是运用理论知识，是远远不够的.我们必须理论联系实际，才能很好的将《Oracle 12C数据库基础教程》这门课程学好，并用于实际案例中，能够在多学科背景下的团队中承担个体、团队成员以及负责人的角色，还能够设计针对复杂工程问题的解决方案，设计满足特定需求的系统、单元(部件)或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素。同时，经过这次课程设计，：能够基于科学原理并采用科学方法对复杂工程问题进行研究，包括设计实验、分析与解释数据、并通过信息综合得到合理有效的结论，我发觉自己的动手操作能力变强了，具有自主学习和终身学习的意识，有不断学习和适应发展的能力，也为自己对以后的工作充满信心和希望打下了良好的基础。